MSWordSST – A New Steganalytical Tool for Microsoft Word Documents

Ivan Stojanov, Aleksandra Mileva, Done Stojanov and Natasa Stojkovik

Faculty of Computer Science
University "Goce Delčev", Štip
Republic of Macedonia
ivan.stojanov1990@gmail.com,
{aleksandra.mileva, done.stojanov, natasa.stojkovik}@ugd.edu.mk

Abstract. Text steganography is the art of hiding data in the text messages or text documents, so that no one suspects it exists. As opposite, the basic task of steganalysis is to detect which carriers have secret bits encoded in them. In this paper, we present a new software tool for steganalysis of nine different format-based steganographic methods in MS Word documents. First, we describe the detection features that we use, which are extracted from the examined document, and after that, we perform experiments for performance analysis of the tool on the legitimate documents and generated stego documents obtained by four property coding methods.

Keywords: Text steganalysis · Open space method · Property coding · Character coding · Invisible characters

1 Introduction

Steganography is the art of concealing a secret message into some legitimate carrier by its undetectable alteration. According to the type of the carrier, there are: digital media steganography, filesystem steganography, network steganography, text steganography, etc. The text is one of the people's most frequently used and one of the oldest mediums for exchanging information, thus making it very interesting as a stego carrier. So, text steganography uses different text messages or text files as a carrier. Text steganography methods can be classified into three main categories [5]: format-based or structural methods (e.g., line shifting and word shifting methods [16]), linguistic or natural language processing-based methods (e.g., lexical method [28]), and random and statistical generation (e.g., Markov chain-based methods [17] and deep learning-based methods [32]). Format-based or structural methods modify the layout features of existing text to conceal the secret message, without altering the words or sentences. Linguistic or natural language processing-based methods involve manipulation of some lexical, syntactic or semantic features of the text content. When random and statistical generation methods are used for hiding a message, a new text is generated which tries to simulate some features of the normal text. They differ from the other two categories in the way that they do not have a carrier in advance, but have to generate automatically a stego carrier.

The science of detecting whether a given carrier has hidden message in it, is called a steganalysis. Particularly, the text steganalysis tries to identify whether a given text message or text file has hidden information in it, and, if possible, to destroy, modify, extract or recover the secret message. The research about text steganography and its counter fighting is very important, because from one side, hiding data in text documents can be abused by cybercriminals and terrorists, while from the other side, it can have a legal application in copyright protection, content authentication, document tracking, etc. There are many surveys of text steganographic and text steganalysis methods (e.g., [1]).

One of the most popular software for word processing, creating and editing text documents is the Microsoft Word. It supports many advanced and easy to use text formatting features. In

this paper, we present a new steganalytical tool MSWordSST[1], specially designed for MS Word documents, which targets nine format-based steganography methods.

The main contributions of this paper can be summarized as follows:

- we design and implement a new tool for steganalysis of nine format-based methods applied in the MS Word documents
- we perform a performance analysis of the tool on the legitimate documents and generated stego documents, obtained by four property coding methods.

Following the presentation of the related steganalytical work, the rest of the paper is structured as follows. Sect. 2 describes many known format-based steganographic methods for text documents in a categorized manner, with nine of them targeted by the new tool. Sect. 3 describes the features extracted from the examined document, used for steganalytical purpose. Details about the implementation and experimental results of the performance analysis are presented in the Sect. 4. The concluding remarks of this paper are made in Sect. 5.

1.1 Related Work

The main task of most text steganalytical methods is to find the differences in the distribution of specific statistical features between legitimate text carriers and stego carriers. Additionally, different kind of steganalytical attacks can be performed, like visual, structural or format-based, and statistical/probabilistic attacks [1]. Visual attacks, a.k.a. Manipulation by Readers -MBR use human perceptual observation for identifying stego carriers, and if the attacker has access to the stego carrier, he/she can even modify or destroy the hidden message. The observation can be made on the base of syntactic modifications, semantic paraphrasing, lexical, rhetorical changes, etc. Structural or format-based attack modifies the layout features of the stego carrier, with the purpose to modify or destroy the hidden information. Statistical/probabilistic attacker tries to decode or guess the secret message, by using some probability distribution functions. According to the range of the application, there are specific methods, that address particular steganographic technique and universal methods, that try to thwart all steganographic techniques. While the first ones try to achieve higher detection accuracy in practice, the second ones are more attractive, because they do not depend on the embedding algorithm, and they can be applied even to unknown techniques.

Detection of the application of linguistic steganographic methods may involve: use of statistical characteristics of the correlations between the general service words gathered in a dictionary for classification of the given text segments into stego-text segments and normal text segments [10], use of statistical "meta" features for text representation, together with the immune clone mechanism for selection of appropriate features and building effective detectors [30], use of perplexity of normal text and stego-text with the Statistical Language Model [18], use of Bayesian Estimation and Correlation Coefficient methodologies [23], etc.

There are also some specific steganalytical methods that address different subsets of linguistic steganography, like the use of language models and support vector machines (SVMs) [26] for detection of lexical linguistic methods, the use of word embedding feature to detect the semantic distortion when synonym substitution is used [36] or the use of context clusters to estimate the context fitness for synonym substitution [11], etc.

Xiang et al. [29] suggested a steganalytical method specific for feature coding (or character coding), which analyzes the font attributes for each character, and then for each font attribute a characteristics vector is created. This method relays on using a SVM classifier for detecting

[1] https://github.com/ivan0071/SteganalysisApp

the existence of hidden information. Li et al. [13] suggested a steganalytical method specific for word shift coding, based on analysis of text space's neighbor difference.

Most of the steganalysis methods deployed for random and statistical generation steganography use convolutional neural networks [27, 31], bidirectional recurrent neural network [33], convolutional sliding windows (TS-CSW) [35] or semantic correlations between words together with a softmax classifier [34], for extracting high-level semantic or word correlation features of texts, and finding the subtle distribution differences in the semantic space before and after embedding the secret information, etc.

2 Format-based Steganography of MS-Word Documents

Format-based or structural steganographic methods can be divided in several subgroups, listed bellow. Representatives presented here are specifically designed for MS-Word documents, or can be applied to MS-Word documents also.

Line Shift Coding This method is using the position of lines in the text as a carrier of the secret bits [16]. The lines in the text document are shifted vertically (either up or down) by a small amount (e.g., up to 1/300 inch) in a way that this change is not easily visually detectable by the user. Shifting up can represent binary one, shifting down can represent a binary zero and vice versa. The control mechanism for detection of line shifting is the static position of the odd line - only even lines are shifted. So, there is no need of the original document for decoding, and even more, text document as a stego carrier can be in both, electronic and printed formats.

Word Shift Coding This method is using the position of words in the text as a carrier of the secret bits [16]. The words in the text document are shifted horizontally (either left or right) by a small amount (e.g., up to 1/150 inch) in a way that this change is not visually detectable by the user. Similar to the line shift coding, the static position of odd words serves as a control mechanism for determining the shifting of the even words. Low et al. [16] also used a combination of line and word shifting, in a way, that each even line is divided into three blocks, and the middle block is shifted either left or right.

Open Method This group of methods hides data by inserting some special characters into text. The basic open space method add one or two white spaces (binary zero or one) between words, at the end of the sentence, line [4] or paragraph [2]. Similar one can use widening, shrinking or unchanging an inter-word space in the text [9, 14]. Other methods in this group use: Unicode space characters [21], their combination with Zero-Width Character (U+200B) [19, 3], etc. Another method uses four special characters, not visible for the reader in MS Word: Right remark (U+200E), Left remark (U+200F), Zero width joiner (U+200D), and Zero width non-joiner (U+200C) [20]. Combining their presence/absence in a particular order, can result in 16 different combinations, meaning that between each two characters in the document, one can hide four bits.

Character (Feature) Coding This method is altering the features of the characters in the text for hiding bits. First examples [7] include: elongating or shortening the ending part of specific characters, such as h, d. b; changing the size of the dot in the characters, such as i, j (this can be applied for 14 letters in Arabic alphabet also [24]); extending or shortening of the horizontal line in the letter t, etc.

The invisible character method [12] is embedding the secret message in the color of invisible characters (white spaces, tabs, new lines) in an RGB format. Each character can carry up to 24 bits with this method.

Another feature that can be changed in characters for hiding data is the font type. The method Similar English Font Types (SEFT) [6] deploys similar English font types. In order the method not to be detectable, only the capital letters in the cover text are used as carriers of the bits (by changing their font types). This method first selects three different similar fonts (e.g., Century751 BT, CenturyOldStyle, CenturyExpdBT), and then, 26 letters and the space character are coded as triples of capital letters in the selected fonts.

For MS Word documents, the authors of [25] present two more character coding methods: character underline and character scale. The character underline is adding invisible underline styles to the characters, so, each character can carry 8 secret bits. However, some of the characters such as g, j, p, q and y, are excluded from this process, since the underline styles will be noticeable when using it on them. The default text character scale for MS Word documents is 100%. One can hide 1 bit per character, by using 99% scale (e.g., binary one) or 101% (e.g., binary zero) by using the character scale method. Furthermore, by using four different scales, one can hide 2 bits per character. Additional alternative to this method, is to scale every word instead of every character.

Another method that belongs in this group is the method based on Unicode of characters in Multilingual [22]. It uses the fact that 13 letters of the English alphabet appears in another languages with different Unicodes.

Property Coding This method utilizes the properties of document objects other than characters (e.g., paragraph borders) as a carrier of secret information. Two different methods are presented in [25]: paragraph borders and sentence borders. The paragraph borders method is using the ability to add the left and right borders to the paragraph and to colorize them with the colors represented by (R, G, B) components, where $R, G, B > 249$ cannot be distinguished from the white color $(255, 255, 255)$ for the human eye. There are 216 different possibilities for color, which can be applied to the paragraph borders, therefore the color itself can carry 7 bits per border. Additionally, there are 22 paragraph styles which can be used for this method with different number of border widths, and this gives the ability to hide additional 7 bits per the combination style/width. So, one can hide up to 28 bits per paragraph.

The sentence borders method uses the ability to add left and right borders to the sentence with colors not noticeable by the user. Its capacity is hiding 10 bits per border or 20 bits per sentence, where the style is carrying 3 bits per border and the color is carrying 7 bits per border.

The change tracking method deploys change tracking facilities of MS Word documents [15] for collaborative writing. The idea is first, to degenerate the content of the carrier by using common spelling mistakes and typos, mimicking to be the work of an author with inferior writing skills. Next, stego document is produced from the previous text by revising it, with the changes being tracked, making it appear as if another author is correcting the errors. The secret bits are embedded using Huffman coding, in a way that shorter Huffman codes are assigned to degenerations with higher probabilities of occurrences, and longer to degenerations with lower probabilities of occurrences.

Starting from MS-Office 2007, Microsoft introduced the Office Open XML (OOXML) file format, together with the Document Inspector feature, which is used for fast identification and removal of any sensitive, hidden and personal information. Four new steganographic methods for MS-Word, that resist the Document Inspector analysis, are presented in [8]. They include methods with: different compression algorithms, revision identifier values, zero dimension image and macro.

3 Detection Mechanisms

Our new statistical steganalytical tool, specially designed for the MS Word documents, can be seen as a set of specific steganalytical methods that target a subset of the format-based methods presented in the previous section. Since different methods use different entities as a carrier, our tool traverses the document and determines several different features.

	1 page	**10 pages**
Legitimate document	60.322	1002.993
Stego document with CS with 5 stego chars	62.333	1058.471
Stego document with CS with 50 stego chars	64.990	1061.845
Stego document with CU with 5 stego chars	61.209	1114.241
Stego document with CU with 50 stego chars	61.779	1243.126
Stego document with PB with 5 stego chars	61.850	1033.993
Stego document with PB with 50 stego chars	n/a	1036.702
Stego document with SB with 5 stego chars	61.389	1152.630
Stego document with SB with 50 stego chars	62.929	1247.927
Average	62.100	1105.772

Table 1. Results for running time (in seconds) for different document sizes

Next, we present the list of methods that our tool support, together with features that are computed for that method:

- Open space method - the total number of sentences in the document, and the number of sentences that have more than one white space at the end; the total number of words in the document, and the number of words that have more than one white space at the end;
- Two methods with special invisible characters - the total number of occurrences of each of the four invisible special characters for the method from [20]: Right remark (U+200E), Left remark (U+200F), Zero width joiner (U+200D), and Zero width non-joiner (U+200C), together with the total number of occurrences of the Zero-Width Character (U+200B) [19, 3];
- Invisible character method - the total number of invisible characters (blank space, tab space, new line), and the number of consecutive color changes of the invisible characters;
- SEFT method – the total number of words with first uppercase letter in the document, the number of different fonts used for first uppercase letters, and the number of consecutive font changes of first uppercase letters;
- Character scale method – the total number of characters for each scale size separately in the document. The scale size of 100% is marked as default in the program;
- Character underline method – the list of all used combinations of underline style/color with the total number of their occurrences in the document (per character). The combination of "single" underline style/ "automatic" black color is marked as default in the program;
- Paragraph borders method – the list of all used combinations of left paragraph border style/colors with the total number of their occurrences in the document (per paragraph), and the list of all used combinations of right paragraph border style/colors with the total number of their occurrences in the document. The combination of "single" underline style/ "automatic" black color is marked as default in the program;
- Sentence borders method - the list of all used combinations of left sentence border styles/colors with the total number of their occurrences in the document, and the list of all used combinations of right sentence border styles colors with the total number of their occurrences in the document (per sentence). The combination of "single" underline style/ "automatic" black color is marked as default in the program.

4 Implementation and Performance Analysis

The new MSWordSST software tool is developed in Visual C# (Classic Desktop Application). The implementation supports analysis of one MS Word document at a time. One can choose to analyze all methods together, or any subset of methods.

In the implementation, the following results indicate the clear absence of the application of the specific method:

- Open space method - Potential = 0, where Potential is the number of sentences that have more than one white space at the end, and the number of words that have more than one white space at the end;
- Two methods with special invisible characters - the total number equals 0 for each of the five invisible special characters;
- Invisible character method - Potential =1, where Potential is the number of successful color changes of invisible characters;
- SEFT method – the number of different font types equals 1;
- Character scale method, character underline method, paragraph borders method, and sentence borders method – only default values are present.

We performed several experiments, as well as performance analysis, for the steganalysis of the four property coding methods, presented in [25]. We used a PC with Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz 2.60GHz processor, 8GB RAM, and 64-bit Windows 10 Home operating system. For the experiments, we use two different sizes of MSWord documents as cover documents - short of one page, and medium of 10 pages. For each cover document, we hide 5 and 50 characters (if it is possible) with the methods "character scale" (CS), "character underline" (CU), "paragraph border" (PB) and "sentence border" (SB) and we end up with 8 different documents for each document size. For each of these 16 documents, we run full steganalysis and we have measured the time needed for the steganalysis to complete. The obtained results are shown in the Table 1, in seconds. We also provide the results for the legitimate documents, without a hidden message.

One can see that examination of two legitimate documents, with 1 and 10 pages, is the fastest. The running time depends of the size of the document. For one page document we need on an average of 62.1 seconds, while for documents with 10 pages we need 1105.772 seconds on average. One can see that the running time depends also from the size of the hidden message, and from the method chosen to hide data in the Word document. The bigger hidden message for the same steganography method and the same size of the stego document, results in the longer time of document examination. One can see that from all four property coding methods, steganalysis of the Word document with character scale method is the longest in the set of documents with 1 page, while with sentence border method needs longest steganalysis time for documents with 10 pages.

All these times can be improved with different implementations that uses parallel and/or concurrent programming.

Two examples of the implemented tool's output can be observed on the Fig. 1 and Fig. 2. Both examples use Word documents of 10 pages, with a following difference: first document is clear, without a stego message, while the second document contains a stego message of 50 characters, hidden with the Paragraph borders method.

5 Conclusion

The recent advances in text steganography raise the need of more steganalytical solutions for counter fighting. In the choise of a specific or universal solution, we have chosen to use a set

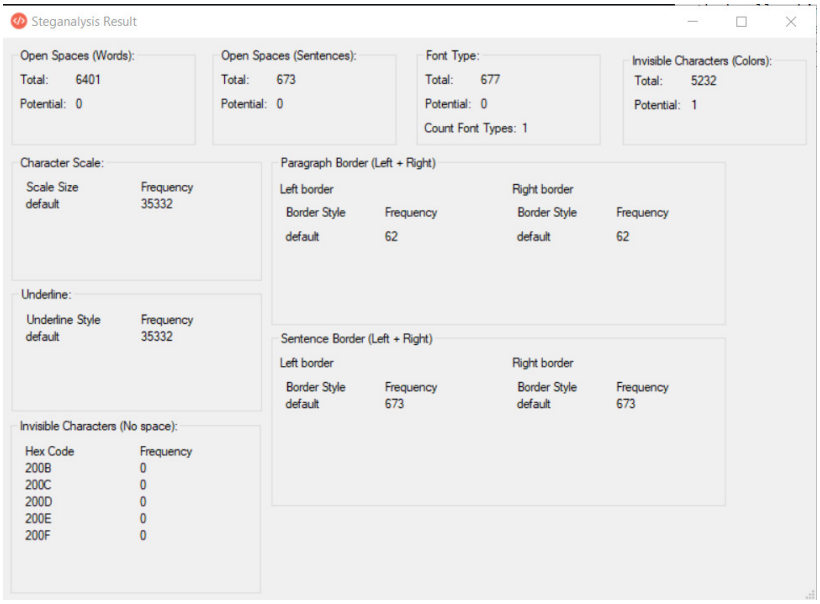

Fig. 1. Output of the MSWordSST tool for a clean document of 10 pages.

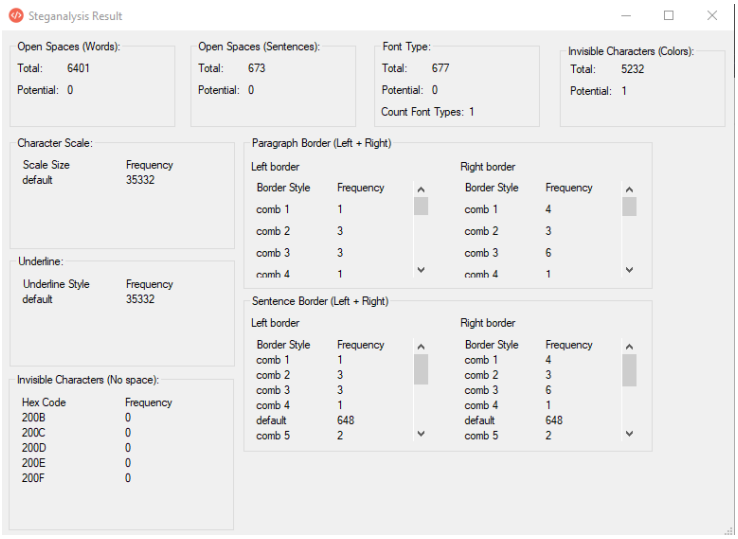

Fig. 2. Output of the MSWordSST tool for a stego document of 10 pages, with 50 characters long secret message, hidden with the Paragraph borders method.

of specific methods that use different document's features, to address several steganography methods on one MS Word document. Currently our implementation covers only nine methods, and its running time can be improved by the use of parallel and/or concurrent programming. For broader covering, additional steganography methods can be also added in the near future..

References

1. Ahvanooey, M. T., Li, Q., Hou, J., Rajput, A. R., Yini, C.: Modern Text Hiding, Text Steganalysis, and Applications: A Comparative Analysis. Entropy 21, 355; doi:10.3390/e21040355 (2019).
2. Alattar, A. M., Alattar, O. M.: Watermarking electronic text documents containing justified paragraphs and irregular line spacing. In Proceedings of the SPIE - Security, Steganography, and Watermarking of Multimedia Contents (2004).
3. Aman, M., Khan, A., Ahmad, B., Kouser, S.: A hybrid text steganography approach utilizing Unicode space characters and zero-width character. Int. J. Inf. Technol. Secur. 9, pp. 85–100 (2017).
4. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Systems Journal, vol. 35, pp. 313–336 (1996).
5. Bennett, K.: Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text. cERIAS Tech Report 2004-13 (2004).
6. Bhaya, W., Rahma, A. M., Al-Nasrawi, D.: Text Steganography based on Font Type in MS-Word Documents. Journal of Computer Science, vol. 9(7), pp. 898–904 (2013).
7. Brassil, J. T. , Low, S., Maxemchuk, N. F.: Copyright protection for the electronic distribution of text documents. In Proceedings of the IEEE, vol. 87 (7), pp. 1181–1196, 1999.
8. Castiglione, A., D'Alessio, B., De Santis, A., Palmieri, F.: New steganographic techniques for the OOXML file format. In Proceedings of the IFIP WG 8.4/8.9 international cross domain conference on Availability, reliability and security for business, enterprise and health information systems, LNCS vol 6908, pp. 344–358 (2011).
9. Chen, C., Wang, S. Z., Zhang, X. P.: Information Hiding in Text Using Typesetting Tools with Stego-Encoding. In Proceedings of the First International Conference on Innovative Computing, Information and Control, pp. 459–462 (2006).
10. Chen, Z., Huang, L., Yu, Z., Yang, W., Li, L., Zheng, X., Zhao, X.: Linguistic Steganography Detection Using Statistical Characteristics of Correlations between Words. In: Solanki K., Sullivan K., Madhow U. (eds) Information Hiding (IH 2008), LNCS vol. 5284. Springer, Berlin, Heidelberg (2008).
11. Chen, Z., Huang, L., Miao,H., Yang, W., Meng, P.: Steganalysis against substitution-based linguistic steganography based on context clusters. Comput. Elect. Eng., vol. 37(6), pp. 1071–1081 (2011).
12. Khairullah, M.: A Novel Text Steganography System Using Font Color of the Invisible Characters in Microsoft Word Documents. Second International Conference on Computer and Electrical Engineering, pp. 482-484 (2009).
13. Li, L., Huang, L., Zhao, X., Yang, W., Chen, Z.: A Statistical Attack on a Kind of Word-Shift Text-Steganography. National High Performance Computing Center of Hefei, pp. 1503-1507 (2008).
14. Lin, I.-C., Hsu, P.-K.: A Data Hiding Scheme on Word Documents using Multiple-base Notation System. In Proceedings of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'10), pp. 31–33 (2010).
15. Liu, T.-Y., Tsai, W.-H.: A New Steganographic Method for Data Hiding in Microsoft Word Documents by a Change Tracking Technique. IEEE Transactions on Information Forensics and Security, vol. 2(1), pp. 24-30 (2007).
16. Low, S. H., Maxemchuk, N. F., Brassil, J. T., O'Gorman, L.: Document marking and identification using both line and word shifting. In Proceedings of the 14th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '95), pp. 853–860 (1995).
17. Luo, Y., Huang, Y., Li, F., Chang, C.: Text steganography based on ci-poetry generation using Markov chain model. KSII Trans. Internet Inf. Syst., vol. 10(9), pp. 4568–4584, Sep. (2016).
18. Meng, P., Hang, L., Yang, W., Chen, Z., Zheng, H.: Linguistic Steganography Detection Algorithm Using Statistical Language Model. In International Conference on Information Technology and Computer Science (2009).
19. Odeh, A., Elleithy, K.: Steganography in Text by Merge ZWC and Space Character. In 28th International Conference on Computers and Their Applications, CATA-2013 (2013).
20. Odeh, A., Elleithy, K., Faezipour, M.: Steganography in Text by Using MS Word Symbols. In Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education (2014).
21. Por, L.Y., Wong, K., Chee, K.O.: UniSpaCh: A text-based data hiding method using Unicode space characters. J. Syst. Softw. 85, pp. 1075–1082 (2012).

22. Rahma, A. M., Bhaya, W., Al-Nasraw, D.i: Text Steganography Based on Unicode of Characters in Multilingual. International Journal of Engineering Research and Applications, vol. 3(4), pp. 1153–1165 (2013).
23. Samanta, S., Dutta, S., Sanyal, G.: A real time text steganalysis by using statistical method. In 2016 IEEE International Conference on Engineering and Technology (ICETECH), pp. 264–268 (2016).
24. Shirali-Shahreza, M., Shirali-Shahreza, S.: A New Approach to Persian/Arabic Text Steganography. In Proceedings of the 5th IEEE/ACIS international Conference on Computer and Information Science and 1st IEEE/ACIS, pp. 310–315 (2006).
25. Stojanov, I., Mileva, A., Stojanovic, I.: A New Property Coding in Text Steganography of Microsoft Word Documents. In Securware 2014: The Eighth International Conference on Emerging Security Information, Systems and Technologies, pp.25 - 30 (2014).
26. Taskiran, C. M., Topkara, M., Delp, E. J.: Attacks on lexical natural language steganography systems. In Proceedings of SPIE - The International Society for Optical Engineering 6072:607209–607209–9 (2006).
27. Wen, J., Zhou , X., Zhong , P., Xue, Y.: Convolutional Neural Network Based Text Steganalysis. IEEE Signal Processing Letters, Vol. 26(3) (2019).
28. Winstein, K.: Lexical steganography through adaptive modulation of the word choice hash. [Online]. Available: http://web.mit. edu/keithw/tlex/ (1998).
29. IXiang, L., Sun, X., Luo, G., Gan, C.: Research on Steganalysis for Text Steganography Based on Font Format. In Third International Symposium on Information Assurance and Security, pp. 490 – 495 (2007).
30. Yang, H., Cao, X.: Linguistic Steganalysis Based on Meta Features and Immune Mechanism. Chinese Journal of Electronics, Vol. 19(4), pp. 661–666 (2010).
31. Yang, Z., Wei, N., Sheng, J., Huang, Y., Zhang, Y.-J.: TS-CNN: Text Steganalysis from Semantic Space Based on Convolutional Neural Network. arXiv:1810.08136 [cs.CR] (2018).
32. Yang, Z., Guo, X., Chen, Z., Huang, Y., Zhang, Y.-J: RNN-stega: Linguistic steganography based on recurrent neural networks. IEEE Trans. Inf. Forensics Secur., vol. 14(5), pp. 1280–1295 (2019).
33. Yang, Z., Wang, K., Li, Huang, Y., Zhang, Y.-J.: TS-RNN: Text Steganalysis Based on Recurrent Neural Networks.' IEEE Signal Processing Letters, Vol. 26(12) (2019).
34. Yang, Z., Huang, Y., Zhang, Y.-J.: A Fast and Efficient Text Steganalysis Method. IEEE Signal Processing Letters, Vol. 26(4) (2019).
35. Yang, Z., Huang, Y., Zhang, Y.-J.: TS-CSW: text steganalysis and hidden capacity estimation based on convolutional sliding windows. Multimedia Tools and Applications (2020).
36. Zuo, X., Hu, H., Zhang, W., Yu, N.: Text Semantic Steganalysis Based on Word Embedding. In: Sun X., Pan Z., Bertino E. (eds) Cloud Computing and Security (ICCCS 2018), LNCS vol.11066, Springer, Cham (2019).